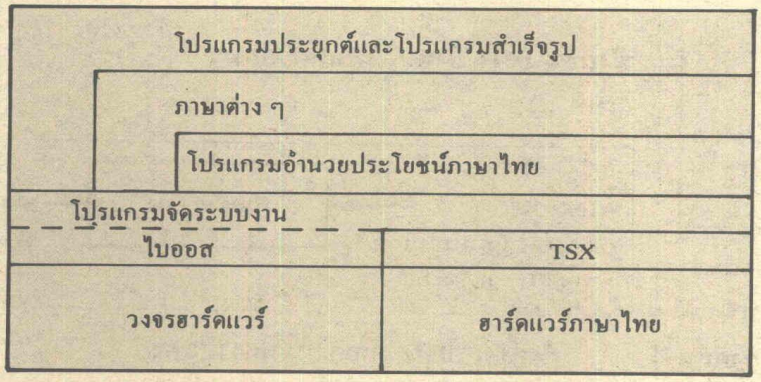
THAI.COM กับ TSX

การพัฒนาระบบภาษาไทยบนเครื่องไมโครคอมพิวเตอร์มีจุดประสงค์เพื่อดัดแปลงโปรแกรมสำเร็จรูปต่าง ๆ ให้สามารถประมวลผลภาษาไทยได้ รวมทั้งเอื้ออำนวยให้มีการสร้างโปรแกรมสำเร็จรูปภาษาไทยขึ้นใช้งานเอง

ห้องปฏิบัติการวิจัยไมโครคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์

วิธีการดัดแปลงนี้เราจึงทำในระดับของโปรแกรมจัดระบบงาน โดยไม่เข้าไปยุ่งเกี่ยวกับโปรแกรมสำเร็จรูปเหล่านั้นโดยตรง ทั้งนี้เพื่อหลีกเลี่ยงความซ้ำซ้อน และผลกระทบข้างเคียงที่ไม่พึงประสงค์ โครงสร้างของระบบไมโครคอมพิวเตอร์ที่ได้รับการดัดแปลงให้สามารถประมวลผลภาษาไทย จึงมีโครงสร้างดังรูปที่ 1 ซึ่งมีการสร้างโปรแกรมจัดระบบงานภาษาไทยในลักษณะของ TSX (Thai System Extensions)



รูปที่ 1 โครงสร้างระบบภาษาไทยบนเครื่องไมโครคอมพิวเตอร์ มีวงจรฮาร์ดแวร์รองรับการทำงานทั้งหมด โปรแกรมต่าง ๆ จะถูกจัดระบบโดยตัวโปรแกรมจัดระบบงาน ตัว TSX จะแฝงอยู่เป็นส่วนหนึ่งของโปรแกรมจัดระบบงานด้วย

TSX คืออะไร

TSX เป็นโปรแกรมที่ถูกสร้างขึ้นมาเพื่อทำงานร่วมกับโปรแกรมจัดระบบงานของไมโครคอมพิวเตอร์ ตัว TSX จะแฝงอยู่เป็นส่วนหนึ่งของโปรแกรมจัดระบบงาน คอยช่วยจัดระบบการใช้ภาษาไทย แนวทางการพัฒนาภาษาไทยด้วยระบบ TSX นี้ทำให้เรา**ไม่จำเป็นต้องแก้ไขวงจรฮาร์ดแวร์**ของเครื่องเลย

TSX ซึ่งมีชื่ออยู่บนแผ่นดิสค์ว่า THAI.COM (บนเอ็มเอสดอส) ยังทำหน้าที่เป็นตัวกลางที่เชื่อมโยงระหว่าง ผู้ใช้กับไบออสของเครื่องอีกด้วย สำหรับโปรแกรม THAI.COM เวอร์ชั่น 3.52 ขึ้นไปจะอนุญาตให้ผู้ใช้ เรียกใช้งานฟังก์ชันต่างๆ สำหรับงานภาษาไทยโดยเฉพาะ การเรียกใช้นี้ทำได้ในระดับของภาษาแอสเซมบลี โดยเรียกใช้ด้วยอินเตอร์รัพท์หมายเลข 69H อย่างไรก็ตามในภาษาระดับสูงบางภาษา ก็สามารถเขียนโมดูลภาษาแอสเซมบลีสั้นๆ เพื่อเรียกใช้ฟังก์ชันนี้ได้เช่นกัน

ฟังก์ชันใน INT 69

ฟังก์ชันใน INT 69 นี้ ประกอบด้วยฟังก์ชันการใช้งานที่เกี่ยวข้องกับการประมวล

ผลภาษาไทยโดยตรง ซึ่งมีทั้งหมด 12 ฟังก์ชัน ดังตารางที่ 1

หมายเลข	การทำงาน
00H	ตรวจสถานะตัวอักษร
01H	ตรวจระดับการแสดงผลบนจอภาพ
02H	ตรวจระดับการแสดงผลบนเครื่องพิมพ์
03H	กำหนดโหมดตัวอักษร
04H	กำหนดระดับการแสดงผลบนจอภาพ
05H	กำหนดจำนวนบรรทัดการแสดงผลบนจอภาพ
06H	กำหนดระดับการพิมพ์ของเครื่องพิมพ์
07H	เลือกชนิดเครื่องพิมพ์
08H	อีนาเบิ้ลหรือดิสเอเบิ้ลฟังก์ชันคีย์
09H	อ่านตัวอักษรจากแป้นพิมพ์
0AH	อ่านตัวอักษรจากแป้นพิมพ์และแสดงผล
0BH	อ่านสายตัวอักษรจากแป้นพิมพ์

ตารางที่ 1 ฟังก์ชันที่มีใน INT 69

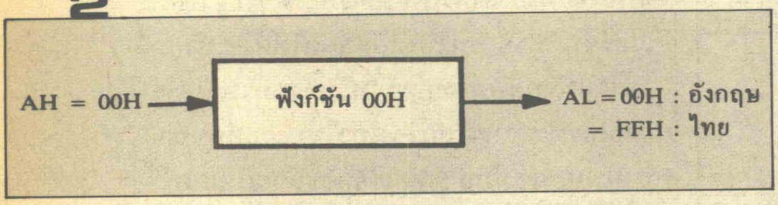
การใช้งานฟังก์ชัน

การใช้งานทุกๆ ฟังก์ชัน จะต้องกำหนดหมายเลขฟังก์ชันที่ต้องการเรียกใช้งานไว้ในรีจิสเตอร์ AH เสมอ และบางฟังก์ชันอาจจำเป็นต้องมีการกำหนดค่าให้กับรีจิสเตอร์ตัวอื่น ๆ ด้วย

หลังจากสิ้นสุดการทำงานแต่ละฟังก์ชันก็จะมีการส่งค่ากลับมาให้ (ในบางฟังก์ชันอาจจะไม่มี) แต่ละฟังก์ชันมีที่ใช้ดังนี้

ฟังก์ชัน 00H ตรวจสถานะตัวอักษร (รูปที่ 2)

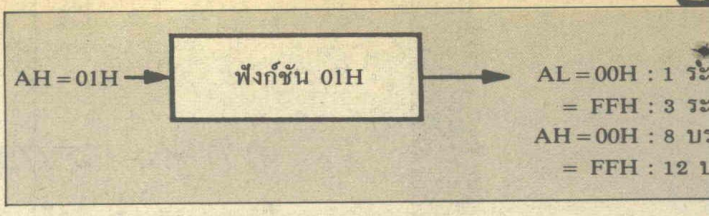
2



ฟังก์ชัน 00H ตรวจสอบสถานะว่าขณะนี้ตัวอักษรที่จะพิมพ์ผ่านแป้นพิมพ์อยู่ในโหมดของภาษาไทยหรือภาษาอังกฤษ โดยส่งค่าที่แสดงโหมดกลับมาในรีจิสเตอร์ AL

ฟังก์ชัน 01H ตรวจระดับการแสดงผล (รูปที่ 3)

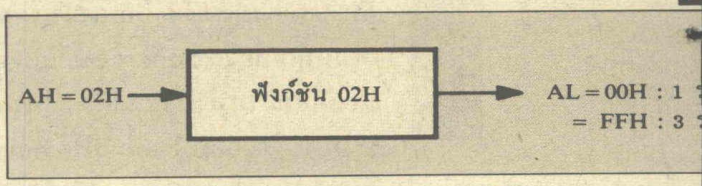
3



ฟังก์ชัน 01H ตรวจสถานะการแสดงผลบนจอภาพ เมื่อมีการแสดงผลตัวอักษรภาษาไทย จะมีการแสดงผลตัวอักษรในบรรทัดเดียวกันหมด (AL = 00H) หรือแสดงผลแบบมีการจัดระดับบนล่างของสระและวรรณยุกต์ (AL = FFH) รวมทั้งจะส่งค่าแสดงจำนวนบรรทัดต่อหนึ่งหน้าจอกลับมาในรีจิสเตอร์ AH ด้วย โดยถ้าเป็นแบบ 8 บรรทัดต่อหน้าจอ AH จะมีค่า 00H และถ้าเป็น 12 บรรทัดต่อหน้าจอ ค่า AH จะเป็น FFH

ฟังก์ชัน 02H ตรวจระดับการแสดงผลของเครื่องพิมพ์ (รูปที่ 4)

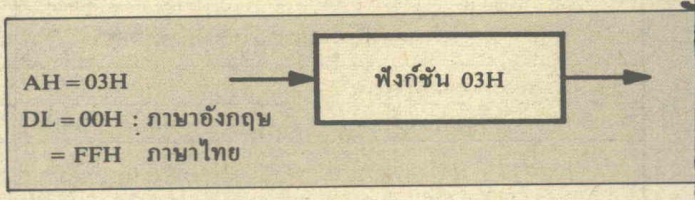
4



ฟังก์ชัน 02H ตรวจสถานะการแสดงผลบนเครื่องพิมพ์ เมื่อมีการพิมพ์ตัวอักษรภาษาไทย จะมีการแสดงผลแบบจัดระดับสระวรรณยุกต์ (AL = FFH) หรือแสดงผลในบรรทัดเดียวกันหมด (AL = 00H)

ฟังก์ชัน 03H กำหนดโหมดตัวอักษร (รูปที่ 5)

5

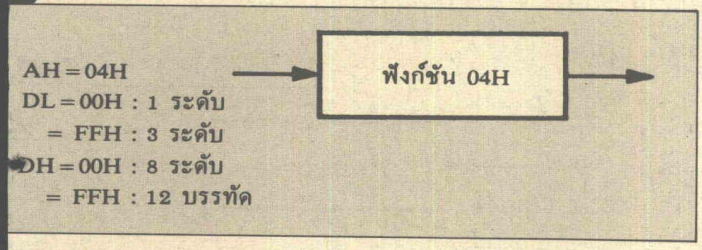


ฟังก์ชัน 03H เลือกโหมดการแสดงผลตัวอักษรบนจอภาพ เมื่อกดอักษรจากแป้นพิมพ์ โดยกำหนดโหมดไว้ในรีจิสเตอร์ DL ถ้า DL = 00H เป็นการเลือกโหมดภาษาอังกฤษ ถ้า DL = FFH เป็นการเลือกโหมดภาษาไทย หลังจากสิ้นสุดการทำงาน

เคอร์เซอร์จะเปลี่ยนรูปร่างไปตามโหมดของตัวอักษร กล่าวคือ เคอร์เซอร์จะมีรูปร่างเป็นขีดเมื่ออยู่ในโหมดภาษาอังกฤษ และมีรูปร่างหนาทึบเมื่ออยู่ในโหมดภาษาไทย ฟังก์ชันนี้จะไม่เปลี่ยนแปลงระดับการแสดงผล และจำนวนบรรทัดบนจอภาพ ตัวอย่างเช่น การแสดงผลเดิมเป็นโหมดภาษาอังกฤษ 1 ระดับ 8 บรรทัด เมื่อเรียกใช้ฟังก์ชันนี้โดยกำหนด DL = FFH จะเป็นการเลือกการแสดงผลภาษาไทย 1 ระดับ 8 บรรทัด ในกรณีที่ค่า DL ไม่เป็น 00H หรือ FFH โปรแกรมจะรีเทอร์นกลับโดยไม่ทำงานใด ๆ ให้

ฟังก์ชัน 04H กำหนดระดับการแสดงผล (รูปที่ 6)

6

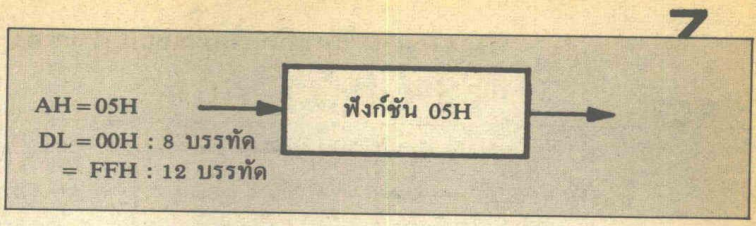


ฟังก์ชัน 04H กำหนดระดับการแสดงผลบนจอภาพ เมื่อมีการแสดงผลภาษาไทยบนจอภาพ จะมีการจัดระดับของสระวรรณยุกต์หรือไม่ ถ้า DL = 00H จะไม่มีการจัดระดับ ถ้า DL = FF จะมีการจัดระดับ รวมทั้งค่าในรีจิสเตอร์ DH จะเป็นตัวกำหนดจำนวนบรรทัดบนจอภาพ ในกรณีที่เรียกใช้ฟังก์ชันนี้ ถ้า DL ไม่เป็นค่า 00H หรือ FFH โปรแกรมจะรีเทอร์นกลับโดยไม่สนใจค่าในรีจิสเตอร์ DH แต่ถ้าค่าในรีจิสเตอร์ DL เป็น 00H หรือ FFH จึงจะทำการกำหนดระดับการแสดงผล แล้วจึงทำการตรวจสอบค่าในรีจิสเตอร์ DH ซึ่งถ้าค่าในรีจิสเตอร์ DH ไม่ใช่ 00H หรือ FFH โปรแกรมก็จะรีเทอร์นกลับไป

ฟังก์ชัน 05H กำหนดจำนวนบรรทัดแสดงผล (รูปที่ 7)

ฟังก์ชัน 05H ใช้ในการกำหนดจำนวนบรรทัดของการแสดงผลบนจอภาพ เมื่ออยู่

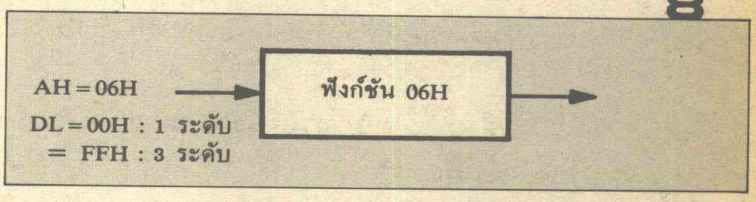
7



ในโหมดของการแสดงผลแบบ 3 ระดับ โดยการกำหนดค่าไว้ใน DL ถ้า DL = 00H การแสดงผลจะเป็นแบบ 8 บรรทัดต่อหน้าจอ และถ้า DL = FFH จะเป็นการแสดงผลแบบ 12 บรรทัดต่อหน้าจอ

ฟังก์ชัน 06H กำหนดระดับการพิมพ์ของเครื่องพิมพ์ (รูปที่ 8)

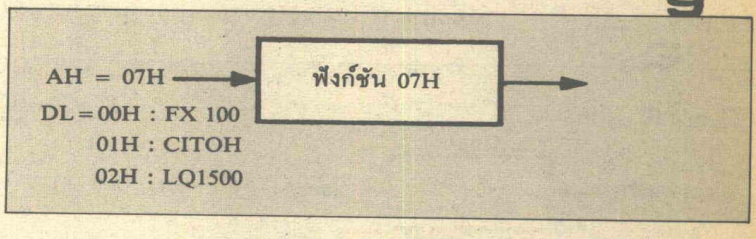
8



ฟังก์ชัน 06H กำหนดระดับการแสดงผลบนเครื่องพิมพ์ ถ้ารีจิสเตอร์ DL มีค่าเป็น 00H จะไม่มีการจัดระดับอักษรภาษาไทย แต่ถ้า DL เป็น FFH จะมีการจัดระดับอักษรภาษาไทย ฟังก์ชันนี้จะทำการกำหนดระยะความห่างของบรรทัด ตามชนิดของเครื่องพิมพ์ที่ติดตั้งอยู่แล้ว

ฟังก์ชัน 07H เลือกชนิดของเครื่องพิมพ์ (รูปที่ 9)

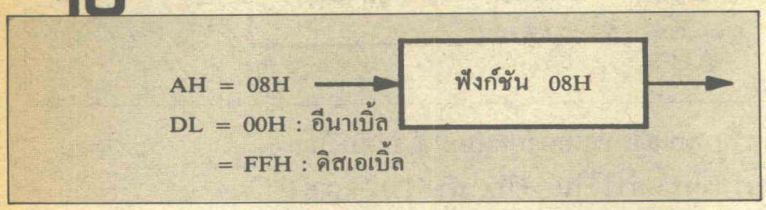
9



ฟังก์ชัน 07H เลือกชนิดของเครื่องพิมพ์โดยกำหนดหมายเลขของเครื่องพิมพ์ไว้ในรีจิสเตอร์ DL ฟังก์ชันนี้จะไม่เปลี่ยนแปลงระดับการแสดงผลที่กำหนดไว้ก่อนหน้านี้ ผลของฟังก์ชันจะกำหนดตัวอักษรของการพิมพ์ ให้เป็นตัวอักษรปกติตามชนิดของเครื่องพิมพ์เสมอ ในกรณีที่ค่า DL ไม่อยู่ในช่วงที่กำหนด ฟังก์ชันนี้จะรีเทอร์นกลับโดยไม่ทำงานใด ๆ ให้

ฟังก์ชัน 08H อีนาเบิ้ลหรือดิสเอเบิ้ลฟังก์ชันคีย์ (รูปที่ 10)

10



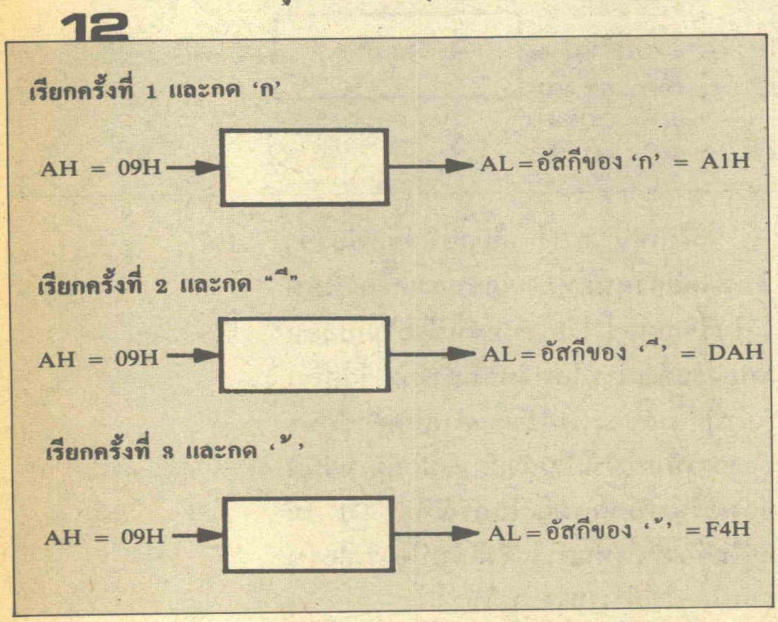
ฟังก์ชัน 08H กำหนดฟังก์ชันคีย์ ALT F1 ถึง ALT F10 ให้อยู่ในลักษณะอีนาเบิ้ลหรือดิสเอเบิ้ล สถานะการใช้งานปกติจะเป็นอีนาเบิ้ล คือใช้ ALT F1-F10 เพื่อเปลี่ยนโหมดไทย–อังกฤษ หรือจัดระดับได้ ส่วนดิสเอเบิ้ล หมายถึงยกเลิกการใช้ ALT F1-F10 ทำให้การกด ALT F1-F10 ไม่มีผลต่อการเปลี่ยนโหมดหรือจัดระดับแต่อย่างไร

ฟังก์ชัน 09H อ่านตัวอักษรจากแป้นพิมพ์ (รูปที่ 11)

11

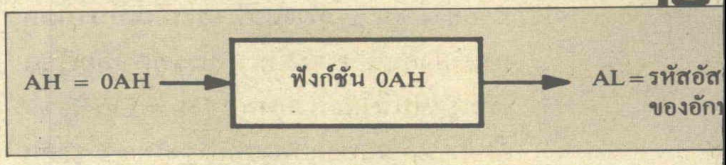
ฟังก์ชัน 09H เป็นฟังก์ชันที่ใช้ในการอ่านตัวอักษรจากแป้นพิมพ์เข้าสู่รีจิสเตอร์ AL และไม่มีการแสดงผล โหมดของตัวอักษร (ไทยหรืออังกฤษ) ขึ้นอยู่กับการกำหนดโหมดก่อนหน้านี้ และในกรณีที่เป็นการกดตัวอักษรภาษาไทย จะมีการผสมสระวรรณยุกต์ให้ด้วย (ไม่ว่าในขณะนั้นจะมีการแสดงผลแบบ 1 ระดับ หรือ 3 ระดับก็ตาม) ยกตัวอย่างเช่น เมื่อเรียกใช้ฟังก์ชัน 09H (รูปที่ 12)

12



ฟังก์ชัน 0AH อ่านตัวอักษรจากแป้นพิมพ์และแสดงผล (รูปที่ 13)

13



ฟังก์ชัน 0AH มีลักษณะการทำงานเหมือนกับฟังก์ชัน 09H แต่หลังจากการกดตัวอักษรแล้ว จะมีการแสดงตัวอักษรนั้นบนจอภาพด้วย การแสดงผลจะมีการจัดระดับสระและวรรณยุกต์ในทันที

ฟังก์ชัน 0BH อ่านสายตัวอักษรจากแป้นพิมพ์ (รูปที่ 14)

14

ฟังก์ชัน 0BH อ่านสายตัวอักษรและแสดงผลอย่างต่อเนื่อง จนกว่าจะกดปุ่มรีเทอร์น ตัวอักษรภาษาไทยจะมีการจัดระดับของสระและวรรณยุกต์ เพื่อการแสดงผลที่ถูกต้อง หลังจากกดปุ่มรีเทอร์น ตัวอักษรทั้งหมดที่พิมพ์จะถูกส่งกลับไปยังแอดเดรสที่ชี้โดยรีจิสเตอร์ DS และ DX จำนวนตัวอักษรที่จะพิมพ์ได้จะรวมกันไม่เกิน 256 ตัวอักษร โดยที่ไบท์แรกของข้อมูลจะบอกความยาวของตัวอักษรที่พิมพ์ ตามด้วยรหัสอัสกีของตัวอักษรที่พิมพ์

ถึงแม้ว่าฟังก์ชันนี้จะทำการแสดงผลแบบ 3 ระดับในทันที แต่ที่จริงแล้วฟังก์ชันนี้จะไม่ทำการเปลี่ยนแปลงระดับของการแสดงผลบนจอภาพก่อนการเรียกใช้ ยกตัวอย่างเช่น ก่อนที่จะมีการเรียกใช้ฟังก์ชัน มีการกำหนดระดับการแสดงผลเป็น 1 ระดับ เมื่อมีการเรียกใช้ และกดรีเทอร์นแล้ว ระดับการแสดงผลก็จะเป็น 1 ระดับเช่นเดิม

ตัวอย่างการเรียกใช้งานฟังก์ชัน

การเรียกใช้ฟังก์ชันภาษาไทยใน INT 69 นี้ จะใช้ภาษาแอสเซมบลีในการเรียกใช้ ตัวอย่างของโปรแกรมภาษาแอสเซมบลี ซึ่งเรียกใช้ INT 69 มีดังรูปที่ 15

15

```
;PROGRAM DISABLE KEY FUNCTION
DISABLE:    MOV AH, 08H    ;FUNCTION 08
            MOV DL, 0FFH   ;DISABLE
            INT  69H       ;CALL ROUTINE
            RET
;PROGRAM READ KEYBOARD BUFFER
READ BUFF: MOV AH, 0BH     ;FUNCTION READ
            MOV DX, 0200H  ;START BUFFER
            INT  69H       ;CALL ROUTINE
            RET
```

ในภาษาระดับสูงบางภาษา สามารถที่จะสร้างข้อมูลของภาษาแอสเซมบลีแทรกเข้าไปได้ เช่น ภาษาปาสคาล ตัวอย่าง โปรแกรมในรูปที่ 16 เป็นการใช้ภาษาปาสคาลที่มีโมดูลของภาษาแอสเซมบลีแทรกอยู่ โปรแกรมภาษาปาสคาลนี้เป็นโปรแกรมตัวอย่างของฟูลสกรีนเอดิเตอร์ ที่ใช้ภาษาแอสเซมบลีทำการเรียกใช้ฟังก์ชัน 09H ในการอ่านตัวอักษรจากแป้นพิมพ์ ประโยชน์ข้อหนึ่งที่เห็นได้ชัดคือ ผู้เขียนภาษาระดับสูง ลดงานเขียนโปรแกรมในด้านการแม็ปแป้นพิมพ์จากภาษาอังกฤษมาเป็นภาษาไทย และการผสมสระวรรณยุกต์ลงไปได้มาก

16 โปรแกรม MICRO.WS แสดงการเรียกใช้ INT 69 ด้วยโมดูลภาษาแอสเซมบลีทำงานร่วมกับปาสคาล

```
Program Micro_WS(input,output);
{-----------------------------------------}
{This program is proposed to demonstrate for}
{using INLINE to invoke INT 69H routine.    }
{The function examples are                  }
{    - function 03 SET_MODE                 }
{    - function 08 KEY_ACT                  }
{    - function 09 READ_KEY                 }
{-----------------------------------------}
Var  key          :char;
     x,y          :integer;
     thai         :boolean; {T=thai F=eng}
Procedure up_char;
begin
  x:=wherex; y:=wherey;
  gotoxy(x-1,y-1);
  write(key);
  gotoxy(x,y);
end;

Procedure norm_char;
begin
  x:=wherex; y:=wherey;
  if x=80 then
     begin
       write(key);
       gotoxy(1,y+3);
     end
   else
     write(key);
end;

Procedure down_char;
begin
  x:=wherex; y:=wherey;
  gotoxy(x-1,y+1);
  write(key);
  gotoxy(x,y);
end;

Procedure read_key;
begin
  inline
    ($B4/$09/      {MOV AH,9        ;FUNCTON READ_KEY}
     $CD/$69/      {INT 69H         ;INTERRUPT EXTENED}
     $A2/key);     {MOV [KEY],AL    ;PUT CHAR TO KEY}
end;

Procedure enable;
begin
  inline
    ($B4/$08/      {MOV AH,8        ;FUNCTON ENABLE}
     $B2/$00/      {MOV DL,00       ;ENABLE FUNCTOIN KEY}
     $CD/$69);     {INT 69H         ;INTERRUPT EXTENED}
end;

Procedure disable;
begin
  inline
    ($B4/$08/      {MOV AH,8        ;FUNCTON ENABLE}
     $B2/$FF/      {MOV DL,FF       ;DISABLE FUNCTOIN KEY}
     $CD/$69);     {INT 69H         ;INTERRUPT EXTENED}
end;
```